

非手続き型言語系に関する研究

著者	富樫 敦
号	945
発行年	1983
URL	http://hdl.handle.net/10097/9681

氏 名	と 富	がし 榎	あつし 敦
授 与 学 位	工	学	博 士
学位授与年月日	昭和 59 年 3 月 27 日		
学位授与の根拠法規	学位規則第 5 条第 1 項		
研究科，専攻の名称	東北大学大学院工学研究科 (博士課程) 電気及通信工学専攻		
学 位 論 文 題 目	非手続き型言語系に関する研究		
指 導 教 官	東北大学教授 野口 正一		
論 文 審 査 委 員	東北大学教授 野口 正一	東北大学教授 木村 正行	
	東北大学教授 伊藤 貴康	東北大学助教授 原尾 政輝	

論 文 内 容 要 旨

プログラミング言語は、人間が計算機と意志疎通を図るための手段であり、計算機を使って情報を処理するための道具である。従って、計算機を有効に利用できるかは、道具である言語によって大きく変わってくる。事実、計算機がいかに速く処理し、いかに大量の情報を記憶できようと、言語が極端に難しかったり、逆に余りにも易しく処理の内容を表現するのに不十分であったりすると、使う側に大きな負担を掛けることになる。言語を使う側から見た場合、問題をいかに解くかというよりもできるだけ問題そのものの記述で処理でき、かつ計算機の構造に密着した表現でなく、より抽象化した表現を使える言語が望ましい。これは、使う人にとって自然であり、計算機の知識なしに使える言語を意味する。

現在、計算を目的とするプログラミング言語は、2 種類に大別できる。1 つは手続き型言語であり、もう一方は、非手続き型言語である。手続き型言語においては、例えば、FORTRAN, ALGOL, PASCAL, PL/1 などに代表されるように、その言語を用いて書かれたプログラムの構文要素は、計算機によって実行される命令の系列であり、その順序や実際に実行されたときの実行順序は、求めるべき結果を計算する上で極めて重要となってくる。以上から、書かれたプログラムの意味を理解するためには、実際に実行してみないと分らない点が多く、理解しづらい。これは、プログラムの意味が、計算機（または、抽象的機械）の実行に大きく依存することによる。また、プログラムの検証や作成に関しても、仕様（通常は自然言語、更にはある論理体系の中で形式化される）とプ

プログラミング言語との構造がはなはだしく異なるため、プログラムの検証やプログラムの合成、作成を著しく困難にする。

一方、非手続き型言語は、数学的な論理体系に基づくプログラミング言語である。そこでは、例えば、関数型言語(λ calculus), 等式(または、書き換え)型言語、現実のシステムとしては pure LISP や Prolog に見られるように、プログラムの構成要素が、あることが成り立つことを主張した等式や理論式の形態をとるため、プログラムを実際に行なわずとも理解できる。これは、この言語で書かれたプログラムの意味を、計算機の実行と切り離して独立に表示できることを意味する。また、プログラムとその仕様を与える仕様記述言語が、比較的似かよった形式(あるいは、全く同じ)を取るため、プログラムの合成や検証が比較的容易にできる可能性がある。

このように両者を比較してみると、非手続き型言語には手続き型言語にない多くの長所や利点がある。以上の背景から、数学的概念に準拠した非手続き型言語系について考察するのは、十分意義があり、重要な研究課題である。

現在、非手続き型言語として、1 階述語論理に基盤を置く論理型言語や等式、書き換え型計算モデルに基づく等式型(書き換え型)言語が注目を集めている。論理型言語、中でも Prolog は、Horn 節と入力導出に基礎を置く言語である。Prolog や等式型言語は、パターン照合やバックトラック等、従来人工知能の分野で取り扱われてきたメカニズムにその計算機能の基礎を置くことから、人工知能向き、知識処理向き言語と言われている。しかし、従来の Prolog の枠組では、述語間の性質を記述し、活用するのに不適當で、等式によるデータタイプの代数的仕様記述法などのデータ抽象化の概念を言語の中に自然に導入することが困難である。また、等式型言語と論理型言語は融合の可能性があるだけでなく、融合する方向にいかねなければならないが、現在のところ明らかにされていない。

本論文では、以上の非手続き型言語系に関する歴史的背景と不備な点やまだ明らかにされていない点を踏まえ、知識ベースを志向した論理型言語を設計するための理論的考察と実際(言語設計と処理系の製作)、及び論理型言語と等式型言語の融合という視点に立ち、次の事項について考察する。

- (1) 論理型言語 Prolog の拡張、並びに拡張に伴う計算機構と意味論。
- (2) 等式型プログラムから論理型プログラムへのプログラム変換。
- (3) 計算とプログラムの不動点における解の存在性との関係。特に、成功する計算と最小不動点。公平な計算と最大不動点。
- (4) 等式型言語の計算モデルに関する計算量の理論からの考察。
- (5) 階層的データ抽象化に基づく論理型言語の設計と処理系。

2 章では、以下の章を展開するのに必要な諸定義、諸概念とそれらに付随する基本的結果について述べる。2.2 節では、単一ソートを多ソートまで拡張した多ソート代数に関する諸定義を与える。2.3 節では、書き換えシステムの一例である項書き換えシステムに関する定義を与え、形式体系との関係について述べる。2.4 節では、不動点理論に関する諸結果を要約する。

3 章では、新しい概念と機能を備えた論理型プログラミング言語を提案し、この言語の特徴をモ

デル理論，不動点理論の立場から特性化する。ここで提案する論理型言語は，従来から提唱されている論理型言語 Prolog を2つの観点から拡張した言語になっている。1つは，求めるべき問題を記述した実行文を負節から妥当式を求めることができるようにクラスタ論理式まで拡張した点である。もう一点は，プログラムの構成要素である確定節を，分配法則や結合法則などが表現できるようにクラスタ式まで拡張した点である。3.2節で，この拡張とその意義について述べる。拡張の経路を踏まえて，3.3節では，プログラムの構成要素はPrologと同じ確定節であるが，実行文をクラスタ論理式まで拡張したHorn型プログラムについて考察する。そこでは，プログラムの意味を推論体系，モデル理論，不動点意味論，計算機構の立場から考察し，証明可能，論理的帰結，最小不動点の要素であること，並びに成功する計算が存在することが全て一致することを示す。3.4節では，実行文だけでなく，プログラムの構成要素まで拡張した一般のプログラムについてモデル論的考察を行なう。また，実行文の計算と同じ枠組で定理証明ができることを示す。前節までは成功する計算結果に対する特性化であるが，3.5節では決して停止しないような計算までも許したもので，実行文の解とそれを求める計算機構との関係について議論する。そこでは，決して停止しない計算も扱うので，通常の有限項を無限項まで拡張し，極限操作ができるようにするため距離を導入し，位相的特性化を行なう。Prolog と本論理型言語の比較を図1に示す。

項 目	Prolog	本論理型言語
プログラム	確定節の集合 φ	クラスタ式の集合 L
実行文	負節： $\neg G$	クラスタ M
計算機構	入力導出	クラスタの簡約機構
計算機構の完全性	$\varphi \cup \{ : \neg G \}$ が充足不能であることと， $: \neg G$ の入力反駁が存在することが同値。	M の解の存在することと， M の一般解を求める成功する計算が存在することが同値。
性質の記述	記述できるが，計算過程で使用不可能。	記述でき，計算過程で使用可能。
プログラミング	再帰的定義によるプログラミング	知識ベースを志向したプログラミング
処理系	比較的容易で，単純	比較的困難で，複雑

図1 Prolog と本論理型言語の比較，検討

4章では，論理型言語にデータ抽象化の概念を導入したり，効率良い計算戦略を取り入れる目的から，等式で記述された等式型プログラムを，本論文で提案する論理型言語で書かれた論理型プログラムに変換する方法について述べる。4.2節では，等式型言語とその言語を用いて書かれたプログラムに関する概念を与え，4.3節で，その変換アルゴリズムとその妥当性に関する理論的根拠を与える。そこでは，任意の等式型プログラムが，同等もしくはそれ以上の計算能力を有する論理型プログラムに変換可能であることを示す。再帰的等式型プログラムに関しては，これと等価なHorn

型プログラムによって完全に模倣できることを証明する。ここで形式化した等式型言語は、従来の書き換え型言語や関数型言語を包含するものであり、本章で提案する変換アルゴリズムは、これらの言語で書かれたプログラムに対しても適用可能となる。4.4節では、変換アルゴリズムの応用として、形式文法に対する言語解析が論理型言語の枠組の中で行なえることを示す。

2.3節で述べる項書き換えシステムは、4.2節で展開するように、等式型プログラミング言語の抽象的な計算モデルとなっている。5章では、計算量の理論の立場から、項書き換えシステムに関する種々の決定問題とその時間計算量について論じる。5.2節では、計算量に関する2, 3の定義を与え、導出関係のモデル論的特性化を5.3節で論じる。この2節で、以下の節で議論を展開するための準備を行なう。5.4節では、定理証明を更に一般化した問題として、項書き換えシステム R 、項の順序対 (t, s) が与えられたとき、 R に属する書き換え規則を適用して t から s が導出できるかどうかを判定する導出可能性問題について考察する。5.5節では、無限書き換え系列が存在するかどうかを問う有限停止性問題について、グラフ理論及び計算量の理論の立場から考察する。近年、停止性を判定する方法がいくつか提案されているが、いずれの方法も十分条件として与えられ、可解なクラスを得るに至っていない。この節での議論は、停止性に関して可解な項書き換えシステムのクラスを与えると共に計算量の立場から扱った考察であり、理論的立場からも興味深い。5.6節では、前節までのアルゴリズムを用いた項書き換えシステムに関するその他の問題として、同値性問題と項に関する停止性問題について議論する。これらの問題は、一般の場合は勿論のこと、1つの変数と2つの1変数関数記号しか存在しないような項書き換えシステムの部分クラスに制限した場合も、一般に非可解となることを示す。一方、変数を含まない定数型の項書き換えシステムのクラスに対しては、全て可解となり、多項式時間で決定可能であることを示す。特に、導出可能性問題、同値性問題は、 P 完全となることを導く。この章の結果は、図2のようにまとめることができる。これらの可解性を示すために、両立的閉包、及び強い意味の両立的閉包という概念を導入し、実際に項を書き換えなくても決定できることを示す。ここで与えた決定手続きは、従来言語理論やオートマトンの理論の中に出て来るものとは全く異なるものであり、その有効性、新規性が示される。

決定問題	一般の場合	定数型の場合
導出可能性問題	決定不可能	多項式時間完全
同値性問題	決定不可能	多項式時間完全
項に関する停止性問題	決定不可能	多項式時間決定可能
停止性問題	決定不可能	多項式時間決定可能

図2 5章の結果のまとめ

6章では、3章で与えたプログラムの拡張により、等式を用いて抽象データ・タイプの仕様を記述したように、論理型言語でもデータ・タイプの仕様を記述することができることを示す。更に仕様の階層性、並びに仕様記述に基づくプログラミングについて述べる。論理型言語のこのような性質から、知識ベースを志向したプログラミングを可能にする。

7章では、3章、6章で与えた理論的考察を基盤にして、論理型プログラミング言語の設計ならびにACOS-6/TSS上で“EPICS LISP”を用いて試作した言語の処理系について述べる。

8章は結論であり、本論文で得られた結果の要約、検討と今後残された問題について概説する。

以上、本論文では、知的情報処理を志向した論理型言語を設計するための基礎理論の確立と処理系の実現、及びそれに附随する諸問題について論じた。最後に、Prologにない本論理型言語の特徴とその応用分野を次に記す。

- (1) 実行文の解を求める計算機構と全く同じ枠組で定理証明ができる。
- (2) 従来知的処理システムとして議論されてきた種々のシステム、例えばプロダクションシステム、を本論理型言語で扱うことができる。
- (3) (1)の特徴を活かし、プログラム変換、正当性の証明、プログラム合成を容易にする可能性がある。
- (4) プログラム方法論の立場から重視されているデータ抽象化、モジュール化の概念を反映したプログラミングを可能にする。
- (5) その他の非手続き型言語、例えば等式型言語、との融合を可能にする。

審 査 結 果 の 要 旨

情報化社会の急速な発展に伴い、新しい情報システムの要求が急速にたかまってきた。すなわち従来の数値計算を主体とした計算機システムのみでなく、人間の知的処理も代行できる高度の処理システムの要求である。しかしながらこのシステムを構築するためには従来の情報処理システム設計法と異なる新しいシステム構成法の研究が必要となる。この中でも特に人間の知識を正確に表現し、この知識を活用して複雑な知的処理を効率よく実行させるような言語体系とその処理系の開発はとりわけ重要な研究課題である。著者はこのような立場に立って、数理論理学の体系を基盤とした非手続き型プログラミング言語の研究を行い、今後の知識情報処理システム構築のための基礎的研究を行った。本論文はこれらの成果をまとめたもので全編8章より成る。

第1章は、序論である。

第2章では、本論文の基礎となる諸概念、諸定義を述べ、これから導かれる基本的な結果をまとめている。

第3章では、知識処理のため非手続き型の新しい論理型プログラミング言語を提案し、この言語の性質を詳細に考察している。まず本言語が従来の論理型プログラミング言語 Prolog を二つの立場から自然に拡張したものであることを示し、この言語のもつ基本的性質、すなわちこの言語系の計算機構の完全性、この系における解の存在条件、プログラムの意味の記述等を明確に与えている。ついで本言語系を用いることにより、与えられた問題を解く手続きが自然に記述できることと、その処理系の実現法を与えている。これらは有用な知見である。

第4章では、従来より研究されている等式型プログラムと論理型プログラムの関係について考察し、任意の等式プログラムが論理型プログラムに変換可能であることを具体的なアルゴリズムを与えて示している。なお、再帰的な等式型プログラムは再帰的構造を持った論理型プログラムに変換できることを示している。

第5章では、等式型プログラミング言語を抽象化した項書き換えシステムについて考察し、代表的な決定問題の決定可能性および計算の複雑性について研究を行っている。

第6章では、第3章で提案した論理型プログラミング言語を用いて、抽象データ・タイプの仕様記述法並びに仕様記述に基づくプログラミングの方法について論じている。

第7章では、LISPを用いて構成した言語処理系について述べている。

第8章は結論である。

以上要するに、本論文は今後の情報工学の重要な研究課題である知識情報処理システム構成法について考察し、その基礎となる論理型プログラミング言語について研究を行ったもので、計算機工学、情報工学の発展に寄与するところが少なくない。

よって、本論文は工学博士の学位論文として合格と認める。